

What is Claimed is:

1. A computing system comprising:

5 a CPU communicatively connected to a memory controller being communicatively connected to a memory connected either to an external operating system or to an internal operating system, through a switching unit;

an input unit that is able to input a request of switching;

a first status thereof defined by states of all variable registers thereof when said computing system is only connected to said internal operating system;

10 a second status thereof defined by states of said all variable registers when said computing system is only connected to said external operating system;

15 a switching unit being communicatively connected to said CPU and all said switches, respectively, has backups of both said statuses, and is able to backup a current status, control said general switch to control all said switches to interrupt all serving programs, then load the other status other than said current status to the computing system.

20 2. The computing system as claimed in claim 1 wherein said CPU also communicatively connected to a video memory controller, or a hard disk controller, or a network adapter controller, each of which being respectively and communicatively connected to a video memory, a hard disk, and a network unit connected either to an external network or to an internal network, respectively through a video memory switch, a hard drive switch, and a network switch.

3. The computing system as claimed in claim 2, wherein said input unit comprises a keyboard, or a mouse, or a touch screen, or E-mail, or other information receiver.

25 4. The computing system as claimed in claim 1, wherein a switching unit having a trigger that is able to generate a non-maskable interrupt to said CPU after said

switching unit receives a command of switching from said CPU responding to said request of switching.

5           5.    The computing system as claimed in claim 4, wherein a switch program is kept in said switching unit, and which has backups of both said statuses, and which, after said CPU receives said non-maskable interrupt, is able to backup a current status, control said general switch to control all said switches to interrupt all serving programs, then load the other status other than said current status to the computing system, and finally control said switching unit to reset said trigger.

10           6.    The computing system as claimed in claim 5, wherein said computing system includes a supervising unit being able to monitor an execution process of said switch program running in said CPU, and allow said switch program to proceed if said supervising unit confirms said process has always been in a script of said switch program, or prohibit said switch control unit from any action if said supervising unit fails to confirm.

15           7.    The computing system as claimed in claim 5, wherein said switch program has an ID verification unit thereof used to confirm a validity of a user, and making said switching unit either to proceed or stop respectively when identification is passed or failed.

20           8.    The computing system as claimed in claim 1, wherein said computing system is a computer, or a Personal Digital Assistant, or a mobile phone.

            9.    The computing system as claimed in claim 1, wherein said backups of both said status are kept in said hard disk.

            10.   The computing system as claimed in claim 1, wherein said backups of both said status are kept in said memory.

25           11.   The computing system as claimed in claim 1, wherein said backups of both said status are kept in a network server.

12. The computing system as claimed in claim 1, wherein said computing system is absolutely disconnected from any network either physically or with software whenever desired by a user, no matter which status said computing system is currently in.

13. The computing system as claimed in claim 1, wherein said switches are  
5 electronic switches.

14. The computing system as claimed in claim 1, wherein said switches are mechanical switches.

15. A method of safely and quickly switching between an internal operating system and an external operating system for a computing system having a CPU, an input  
10 unit, a first status, a second status, a switching unit, and, a switch program, wherein said method comprises following steps:

(1) input a request of switching via said input unit when said computing system is in one of said two statuses;

(2) after receiving said request of switching, said CPU runs said switch  
15 program; and

(3) said switch program backs up a current status, control said switching unit to interrupt all serving programs and loads the other status other than said current status to the computing system.

16. The method as claimed in claim 15, said switching unit having a trigger,  
20 and, a switch program kept in said switching unit, wherein step (2) may be

(2.1) after receiving said request of switching, said CPU sends a command of switching to said switching unit;

(2.2) said switching unit sets said trigger which generates a consequent non-maskable interrupt back to said CPU; and

25 (2.3) after and only after said receives said non-maskable interrupt, said CPU runs said switch program.

17. The method as claimed in claim 15, wherein a further sub-step (3.1) may be added into step (3) wherein before said switch program starts backing up said current status, a further confirmation of a judging unit kept in said switching unit is required.

18. The method as claimed in claim 15, wherein in step (1) a request of  
5 switching may be E-mail, keyboard, mouse, command, other information.

19. The method as claimed in claim 15, wherein step (3) is that said supervising unit monitors an execution process of said switch program running in said CPU, and either (3a) allows said switch program to back up a current status, control said switching unit to interrupt all serving programs and loads the other status other than said  
10 current status to said computing system, and finally control said switching unit to reset said first trigger if said supervising unit confirms that said process has always been in a script of said switch program, or (3b) prohibit said switch control unit from any action if said supervising unit fails to confirm.